## What is claimed is:

	\	1
16	$\sqrt{2}$ , $\sqrt{1}$ .	A method comprising:
2		storing first tuples in a first table in a database system;
3		storing second tuples in a second table in the database system;
4		partitioning the first and second tuples into plural portions;
5		redistributing the first and second tuples to plural nodes according
6	to the partiti	oning; and
7		hash joining the first and second tuples to produce result tuples as
8	the first and	second tuples are being redistributed to the plural nodes.
1	2.	The method of claim 1, further comprising:
2		retrieving the result tuples once the hash join is performed.
1	3.	The method of claim 1, further comprising:
2		retrieving the result tuples at random.
1	4.	The method of claim 1, hash joining the first and second tuples to
2	produce resu	ult tuples as the first and second tuples are being redistributed to the
3	plural nodes	further comprising:
4		producing result tuples at one of the plural nodes; and
5		simultaneously producing result tuples at a second of the plural
6	nodes	<b>/</b> .
	./	· ·

1	5.	The method of claim 1, wherein redistributing the first and second
2	tuples to plu	ural nodes comprises redistributing based on split vectors containing
3	predefined r	ranges.
1	6.	The method of claim 5, wherein partitioning the first and second
2	tuples into plural portions comprises:	
3		partitioning first and second tuples into hash tables in each node.
1	7.	The method of claim 6, wherein hash joining the first and second
2	tuples comprises:	
3		allocating a portion of a memory to a first hash table;
4		allocating a second portion of the memory to a second hash table;
5	and	
6		hash joining first tuples in the first hash table with second tuples in
.7	the second h	nash table.
1	8.	The method of claim 7, wherein hash joining the first and second
2	tuples comp	rises:
3		determining that the portion of the memory allocated to the first
4	hash table is full;	
5		allocating a stable storage to the first hash table; and
6		storing first tuples in the stable storage.
1	9.	The method of claim 8, further comprising:
2		continuing to store second tuples in the second hash table; and
3	/	hash joining second tuples in the second hash table with first tuples
4	in the first h	ash table.
	,	

1	10.	The method of claim 9, further comprising:
2		determining that the second portion of the memory allocated to the
3	second hash	table is full;
4		allocating a second stable storage to the second hash table;
5		storing second tuples in the second stable storage; and
6		hash joining second tuples in the second stable storage with first
7	tuples in the	e first hash table.
1	11.	The method of claim 10, wherein hash joining the first and second
2	tuples comp	rises:
3		generating a third hash table once all first tuples and second tuples
4	are redistrib	uted to each node;
5		retrieving one of the first/tuples from the stable storage;
6		hash joining the one of the first tuples with tuples in the second
7	hash table; a	and /
8		storing the one of the first tuples in the third hash table.
1	12.	The method of claim 11, further comprising:
2		retrieving one of the second tuples from the second stable storage;
3	and	
4		hash joining the one of the second tuples with tuples in the third
5	hash table.	
1	13.	A system comprising:
2		a processor;
3		a storage; and
4		instructions executable by the processor, for enabling the system
5	to:	
6		store first tuples in a first table;
7		store second tuples in a second table:

8	partition the first and second tuples into plural portions;
9	redistribute the first and second tuples to plural nodes
10	according to the partitioning; and
11	hash join the first and second tuples to produce result tuples
12	as the first and second tuples are being redistributed to the plural nodes.
1	14. The system of claim 13, wherein the result typles are available
2	once the hash join is performed
1	15. The system of claim 13, wherein the result tuples are available at
2	random.
1	16. The system of claim 13, wherein each node comprises a memory,
2	and wherein the instructions further partition the first and second tuples into
3	plural portions by:
4	partitioning first tuples into first hash tables; and
5	partitioning second tuples into second hash tables, wherein
6	the hash tables are in the memory.
1	17. The system of claim 16, wherein the instructions further:
2	allocate a portion of the memory to the first hash table;
3	allocate a second portion of the memory to the second hash table;
4	and
5	hash join first tuples in the first hash table with second tuples in
6	the second hash table.

1	18.	The system of claim 17, wherein the instructions further:
2		determine that the portion of the memory allocated to the first
3	hash table is	s full; and
4		store first tuples in a stable storage.
1	19.	The system of claim 18, wherein the instructions further:
2		continue to store second tuples in the second hash table; and
3		hash join second tuples in the second hash table with first tuples in
4	the first has	h table.
1	20.	The system of claim 19, wherein the instructions further:
2		determine that the second portion of the memory allocated to the
3	second hash	table is full;
4		allocate a second stable storage to the second hash table;
5		store second tuples in the second stable storage; and
6		hash join second tuples in the second stable storage with first
7	tuples in the	e first hash table.
1	21.	The system of claim 20, wherein the instructions further:
2		generate a third hash table once all first tuples and second tuples
3	are redistrib	uted to each node,
4		retrieve one of the first tuples from the stable storage;
5		hash join the one of the first tuples with tuples in the second hash
6	table; and	
7		store the one of the first tuples in the third hash table.
1	22.	The system of claim 21, wherein the instructions further:
2		retrieve one of the second tuples from the second stable storage;
3	and	(

4		hash join the one of the second tuples with tuples in the third hash
5	table.	
1	23.	An article comprising a medium storing instructions for enabling a
2	processor-b	pased system to:
3		store first tuples in a first table in a database system;
4		store second tuples in a second table in the database system;
5		partition the first and second tuples into plural portions;
6		redistribute the first and second tuples to plural nodes according to
7	the partition	ning; and
8		hash join the first and second tuples to produce result tuples as the
9	first and se	cond tuples are being redistributed to the plural nodes.
1	24.	The article of claim 23, further storing instructions for enabling a
2	processor-based system to:	
3		retrieving the result tuples once the hash join is performed.
1	25.	The article of claim 24, further storing instructions for enabling a
2	processor-b	pased system to:
3		redistribute based on split vectors containing predefined ranges.
1	26.	The article of claim 25, further storing instructions for enabling a
2	processor-b	ased system to:
3		partition first and second tuples into hash tables in each node.
1	27.	The article of claim 26, further storing instructions for enabling a
2	processor-b	ased system to:
3		allocate a portion of a memory to a first hash table;
4		allocate a second portion of the memory to a second hash table;
5	and	

6		hash join first tuples in the first hash table with second tuples in
7	the s	econd hash table.
1	28.	The article of claim 27, further storing instructions for enabling a
2	processor-b	ased system to:
3		determine that the portion of the memory allocated to the first
4	hash table i	s full; and
5		store first tuples in a stable storage.
1	29.	The article of claim 28, further storing instructions for enabling a
2	processor-based system to:	
3		continue to store second tuples in the second hash table; and
4		hash join second tuples in the second hash table with first tuples in
5	the first has	sh table.
1	30.	The article of claim 29, further storing instructions for enabling a
2	processor-b	ased system to:
3		determine that the second portion of the memory allocated to the
4	second hash	n table is full;
5		allocate a second stable storage to the second hash table;
6		store second tuples in the second stable storage; and
7		hash join second tuples in the second stable storage with first
8	tuples in the	e first hash table.
1	31.	The article of claim 30, further storing instructions for enabling a
2	processor-b	ased system to:
3	/	generate a third hash table once all first tuples and second tuples
4	are redistrib	outed to each node;
5		retrieve one of the first tuples from the stable storage;
	,	

6		hash join the one of the first tuples with tuples in the second hash
7	table; and	
8		store the one of the first tuples in the third hash table.
1	32.	The article of claim 31, further storing instructions for enabling a
2	processor-ba	ased system to:
3		retrieve one of the second tuples from the second stable storage;
4	and	
5		hash join the one of the second tuples with tuples in the third hash
6	table.	
	<u>/</u> .	

odes